

Communications of the Association for Information Systems

Volume 24

Article 40

6-1-2009

Virtual Organizations as Electronic Services

Levent V. Orman

Cornell University, orman@cornell.edu

Follow this and additional works at: <https://aisel.aisnet.org/cais>

Recommended Citation

Orman, Levent V. (2009) "Virtual Organizations as Electronic Services," *Communications of the Association for Information Systems*: Vol. 24 , Article 40.

DOI: 10.17705/1CAIS.02440

Available at: <https://aisel.aisnet.org/cais/vol24/iss1/40>

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Communications of the Association for Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Communications of the Association for Information Systems



Virtual Organizations as Electronic Services

Levent V. Orman

Cornell University, Ithaca, New York

orman@cornell.edu

Abstract:

Virtual organizations are flexible organizations, dynamically built from components of existing organizations. There are many advantages to virtual organizations, yet they are rare because successful implementation of such a dynamic design requires a significant technology infrastructure and a complex organizational architecture. This article takes a design approach and proposes such a technology infrastructure based on electronic services, formal semantics, and multi-dimensional ontologies. Organizational components are formally described as services, and placed in multi-dimensional ontologies, not only to capture their structural components, but also their organizational goals and audiences. Such multi-dimensional ontologies are not only useful to describe and search for organizational components, but also to efficiently build larger components from existing units. The architecture is extended to include virtual communities, virtual transactions, and virtual social institutions. The impact of such an architecture on transaction efficiency, organizational flexibility and adaptation, power structure, individual privacy, and organizational dissent are discussed.

Keywords: virtual organization, electronic services, service semantics, multi-dimensional ontologies, organizational architecture, virtual communities

Volume 24. Article 40. pp. 701-718. June 2009

I. VIRTUAL ORGANIZATIONS

A virtual organization is a flexible organization, dynamically built from components of existing organizations. It is characterized by quick assembly and disassembly, as well as sharing of components among multiple organizations [Camarintha et al. 2005]. Virtuality is a matter of degree since most organizations utilize some existing resources (such as employees, buildings, or infrastructure) in their design phase, and most organizations share some resources with other organizations (such as the public infrastructure) during their ongoing operations.

The potential range of sharing, however, is quite wide. At one extreme are organizations that exhibit no virtuality, such as families. They are life-long, built from the ground up, and are non-substitutable. At the other extreme are organizations that are highly virtual, such as worldwide terrorist organizations. They often consist of many relatively independent cells that are temporary, fluid in structure and membership, shared among multiple organizations, and interchangeable with each other. They can quickly join forces to form larger organizations for a specific project, and dissolve into smaller independent components after the project is completed. Similarly, some online retail businesses are built entirely from the services of already existing businesses, such as holding inventory, distribution, billing, web site design, and payment processing. Other examples of highly virtual organizations are concert tours and independent movies. They bring together many independent artists and technicians, and the components of many existing firms for a specific project, and release them after the project is completed [Franke 2002].

Virtual organizations are not very common in mainstream business, government, or social institutions. Although they are highly responsive to changes in the environment, they have difficulty following long term plans and strategies. Their dynamic nature allows them to respond quickly to short term changes. However, their independent components, each with their own goals and plans, often have difficulty coordinating, resolving conflicts, and maintaining consistent progress toward a long term goal [Mowshowitz 2003].

In general, virtual organizations present a number of unresolved issues, such as; the difficulty of quickly locating all the relevant components for each organization, communicating the overall goals and plans to each component, combining and coordinating them into a coherent unit, and establishing mutual trust and confidence among them. Because of these unresolved issues, virtual organizations tend to be short term and task oriented, since short term goals of narrowly defined tasks are easier to communicate to new groups. They also tend to be smaller and tend to focus on non-mission critical tasks, since it is easier to trust new groups with these tasks. A terrorist group for example, can utilize a virtual organization for developing individual operations, but not for building a global leadership team to determine its long term planning and strategy. A service business, for example, can utilize a virtual organization for its customer relationship operations, but not for the design of its strategic communication infrastructure and protocols [Malhotra 2000].

To extend the benefits of virtual organizations to large, complex organizations with long term strategic goals would require a flexible architecture [Ren and Lyytinen 2008]. That architecture would have to facilitate multilevel descriptions of components where larger components can be built from smaller components. It would require formal semantics that describe and enforce long term goals and short term tactics unambiguously. Finally, it would need a multilevel trust mechanism in which the lower level transactions are bound by higher level relationships, without the need for establishing trust at every level between every pair of transacting agents. This article proposes such an architecture to design and implement complex virtual organizations, and describes the possible consequences of such an architecture on a wide variety of organizational issues ranging from control and flexibility, to privacy and dissent [Mowshowitz 2003].

II. DESIGN CRITERIA

There are two major, outstanding design problems with virtual organizations:

1. Search problem: deals with the question of how organizational components find each other, and how they communicate their goals, strategies and plans to each other [Orman 2008a]
2. Incentive problem: deals with how the organizational components build trust to ensure that the goals, strategies and plans are shared and supported by all [Franke 2002]



A resolution of these two outstanding issues would allow a high degree of virtuality in organizations. The degree of virtuality in an organization is determined by the size and complexity of the components borrowed from existing organizations during the design phase, and the size and complexity of the components shared with other organizations during the ongoing operations. Virtuality then, is a measure of reliance on component based design [Elfatraty 2007], and ongoing resource sharing [Orman 2008]. As such, virtuality can be encouraged by an architecture that facilitates sharing of design components and operational resources. Consequently, and in light of the two unresolved issues listed above, two architectural features are likely to help in the design and implementation of virtual organizations:

1. Capturing the formal semantics of organizational components in ontologies, and storing them in a searchable form in universal directories can facilitate the search for components. It can also increase the sharability of those components through more precise and detailed descriptions, leading to sharing of larger and more complex components [Hyvönen 2008].
2. Hierarchical type structures, in which higher level components are built from lower level components can increase the size and complexity of components available for sharing. They can also create a trust mechanism where components do not exist in isolation, but their memberships in higher level constructs constrain and regulate them, discouraging short term opportunistic behavior [Erl 2005].

A service architecture is proposed to incorporate these two features into a comprehensive design architecture. The organizational components will be described as services for the internal and external constituencies of the organization. They will be organized into type hierarchies and their descriptions will be expanded into ontologies stored in universal and searchable directories [Sheth et al 2006; Woods 2006]. Such ontologies are useful in capturing the semantics of components to communicate goals, strategies, and plans. Type hierarchies of these ontologies are useful in building larger and more complex components from smaller and simpler components, and describe the components in terms of their simpler constituent components. The high level components are useful in establishing rules of behavior and enforcing them on their constituents. Such regulation is useful in building reliability and trust between components that rarely interact directly since their memberships in higher level components restrain them to predictable behavior through established, long-term relationships among their higher level super-components [Devedzic 2002; Gomez-Perez et al 2002].

III. A SERVICE ARCHITECTURE

An organization can be viewed as a collection of services delivered to internal and external constituencies. Services are different from organizational processes because of their higher level semantics. They encapsulate organizational processes and describe them in terms of their objectives and deliverables rather than their implementation. A radio station as a service, for example, is viewed as a sequence of delivered music which is both its objective and its output. That is different from the view of a radio station as a collection of processes such as selecting, searching, locating, and playing music, which is its implementation [Cherbakov et al 2005; Rust 2004].

Semantics of services is simpler than the semantics of processes. A service captures only the goal and the output of a process, while the internal details are omitted. As such, services can be formally described by ontologies. An ontology may contain descriptive keywords, dictionaries that define those keywords, type hierarchies that place them in a semantic structure, as well as attributes and constraints that describe the values they take. In general, service descriptions require multi-dimensional ontologies, involving not only the structural semantics of services in terms of their components, but also their use semantics such as; the goal of the service, the target audience of the service, and the time, duration, and location of the service. Each of these dimensions involves a hierarchy of types, attributes of those types, and relationships among the type hierarchies, leading to a complex multi-dimensional ontology [Erl 2005; Riki et al 2004].

Consider a travel agency operating as a virtual organization. It would have to aggregate a variety of services such as plane, train, and bus trips, hotel reservations, car rentals, visits to restaurants, museums, beaches, and zoos, to satisfy the travel needs of a single customer. Each of those is a service provided by an independent component of a virtual organization, and they need to be combined dynamically to serve the specific needs of an individual customer. That requires a formal and searchable description of those services. Moreover, different customers may view the same components differently, and use it to satisfy different needs. For example, a museum trip may be an educational, professional, or leisure activity. A European trip may be a vacation, a retirement activity, an educational trip, or a family reunion. The attributes necessary to describe the service would be different for each goal. That requires the formal descriptions to be customer and goal dependent. The structural components comprising a service are also different for each goal, and even for each customer. Clearly, a European vacation on the Mediterranean is likely to include a beach visit component, but not an educational trip to Europe. This close

interaction between virtual organizations and their service components on the one hand, and the nature of their customers and their goals on the other, requires a multi-dimensional ontology to describe.

A European trip, as an example of a service provided by a travel agency, would have to be described by three type hierarchies, representing its structural components, its goal, and its target audience respectively, as shown in Figure 2.1. In addition, attributes need to be attached to each node, and relationships between nodes need to be captured. Then, a specific trip can be searched, described, and shared to build larger services. A specific European trip such as a college history department class trip can be searched by using any of the three type hierarchies: as a historical sightseeing trip in the goal hierarchy, as a college level history class trip in the target audience hierarchy, or as a trip that includes a visit to the historical museum of Athens in the structure hierarchy. Moreover, once found, such a service can be combined with other services provided by the other components of the virtual organization to build compound services for the internal and external constituents of the organization. For example, a student sightseeing trip can be defined as a combination of historical and natural sightseeing. The new bundle can immediately be formed by combining the two existing bundles. The bundle formation is not merely a trivial exercise, but also follows constraints and rules specific to each type of bundle. For example, a combination of two sightseeing bundles typically combines all the sights of both bundles, but does not duplicate hotels in, or the plane trip to the same city..

The type hierarchies are related to each other via type relationships. Figure 1 shows three such relationships. The red relationship establishes that students on sightseeing trips visit cities; the green relationship shows that college students on sightseeing trips visit museums; and the blue relationship states that college history students on history trips visit historical museums.

These are called stereotypical relationships. They are instantiated at lower levels of the hierarchy either by accepting the stereotypical solution, or by starting with the stereotypical solution and incrementally modifying it. There are also attributes and constraints attached to each relationship. They further describe the relationship and the role of each node in the relationship. They also establish the ability of the constituent nodes to substitute for each other in achieving various goals, and the tradeoffs necessary for the substitution.

For example, a sightseeing trip involving sights in a city and a hotel stay would require the sights and the hotel to be in the same city. That would be a constraint attached to the red relationship in Figure 1. Students on sightseeing trips are often limited to cities where there are youth hostels because of budget limitations. The red relationship in Figure 1 requires a youth hostel reservation for a student sightseeing trip. However, a YMCA, which is a sibling of a youth hostel in the type hierarchy can be substituted for it, creating an alternative bundle. The substitution of a YMCA would impose an additional cost to the trip, and that often requires a tradeoff between two constraints attached to student sightseeing trips: budget constraint and comfort constraint. The trade-offs and constraints are attached to the relationships, but the costs are attached to the individual nodes. One can move from node to node by substituting one node for another and create new bundles, while maintaining the constraints attached to the relationship. Moreover, at each node, one can look ahead to all neighbor nodes, compute the changes to all the constraints, and provide all possible movements and their impact on the constraints and tradeoffs. In the case of the red relationship of Figure 1, a YMCA can be proposed as a substitute for a youth hostel in a student sightseeing vacation, and the specific tradeoff is computed between cost and comfort for that particular substitution. If the utility function of a specific student group is known, the substitution can be done automatically rather than listing all possible substitutions and the tradeoffs they present. [Devedzic 2002; Gomez-Perez. 2004; Orman 2008a].

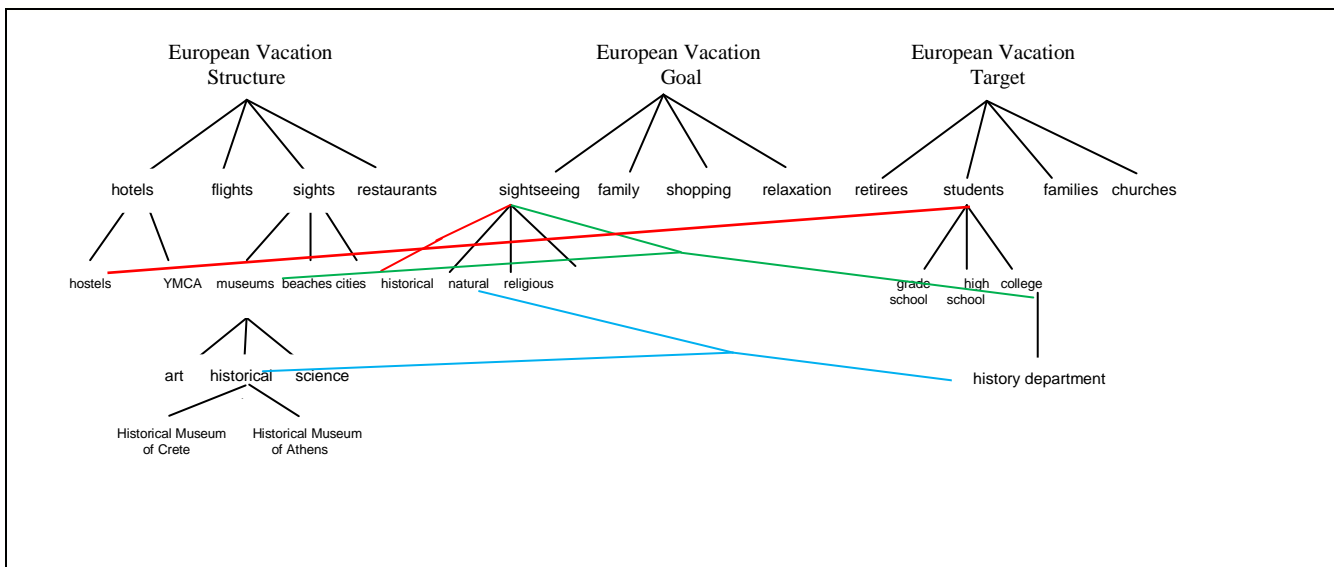


Figure 1. A Segment of a Multidimensional Ontology to Describe the Services of a Travel Agency

Such a multi-dimensional ontology has a number of advantages:

- Services can be described formally and searched electronically. Search for documents is already one of the largest applications on the internet, and it led to the phenomenal success of search businesses through keyword-based document search. Services are more complex than documents both in terms of their structure and their semantics, and search by keywords is rarely an adequate strategy for services. Consider the search for a medical procedure to treat an illness. It requires a complex search involving multiple businesses, specialists, web sites, documents, databases, and reviews, and multiple steps iterating among them until a bundle of resources is located for the specific problem. The search itself is customized for each specific user. It utilizes not only the descriptions of the searched resources, but also the descriptions of users including their symptoms, medical history, and results of diagnostic tests.

Capturing the formal semantics of electronic Web services is already an active research and development area. Languages such as Web Services Description Language (WSDL) capture the semantics of web services using a one-dimensional ontology and a type system based on Extensible Markup Language (XML). Such formal semantics are machine-readable and machine-searchable. The captured formal descriptions are placed in universal directories such the Universal Description, Discovery, and Integration Directory (UDDI) in order to be universally machine-searchable. So far, these efforts have been limited to electronic web services. Organizations contain both electronic and human components, to perform more complex tasks, and achieve more abstract goals. Hence, capturing their semantics requires more complex tools, such as multi-dimensional ontologies, to facilitate both machine and human processing [Alonso 2003; Erl 2005; Sheth et al 2006].

- Organizational components can be described as bundles of services. Bundling can reduce the search effort since only a single search is needed to locate a large number of related and complementary services. However, the reduction in search effort comes at a cost of more complex semantics since aggregates are more difficult to describe. Describing bundles structurally, merely as aggregates of components, is not sufficient since large bundles become increasingly disconnected from the semantics of their components. An effective description would require a multidimensional ontology in which the bundle semantics are captured in abstract attributes, independent of their structural components, as goal and task semantics. Each component of the bundle itself is a bundle; but it is also a service. It can also be described in terms of its abstract attributes or its concrete components, leading to a complex, multidimensional hierarchical structure with the following components [Hearst 2006; Malone 1999]:
 - An abstract semantics for bundles that describes them in terms of their abstract attributes and semantic type hierarchies
 - A structural semantics for bundles that describes them in terms of their concrete components and structural hierarchies

- A relationship between the two hierarchies that links the structural components to the abstract attributes

Consider a search for a medical service such as heart surgery. It could be described by its structural components such as doctors, hospitals, labs, pharmacies, and nursing care. Alternatively, it could be described by its abstract attributes such as cost, success rate, recovery period, fatality rate, repeat surgery risk, and travel distance. The tradeoffs among various abstract attributes would be informative, but they can only be derived by capturing the relationship between those attributes and the structural components such as the choice of doctors, hospitals, and nursing care providers [Orman 2008].

- Organizational components can be combined dynamically to build new and more complex organizational units, facilitating rapid response to opportunities and threats, and allowing the design of experimental organizations. Without formal semantics, any modification to a service component typically requires checking all other components of the bundle for consistency with the new component, and to calculate the impact of the new component on the bundle as a whole. Developing service semantics independent of its structural components allows replacing some components with others without changing the description of the bundle, or extensive consistency checking if the replacement is semantically equivalent to the old component. If a component is replaced with a nonequivalent component, the relationship between structure and semantics allows the calculation of the impact of the replacement on the bundle semantics, as well as the identification of other changes to the bundle necessary to maintain internal consistency. Such a capability would allow dynamic restructuring of services with minimal disruption to the consumers of those services. It would accomplish this by either isolating the consumers from the internal changes, or by describing their impact at the level of bundle semantics that is easily understandable by consumers, and also by maintaining the internal consistency of the services during restructuring. Such a capability requires a complex semantic system imposed on services such as:
 - Semantic description of structural components which would allow their reuse within multiple bundles
 - Semantic description of bundles independent of their structure, which would allow changing their structure by substituting equivalent components for each other, without affecting the bundle semantics
 - A semantic relationship between bundles and components so that the impact of any changes to a component can be calculated and reported at the bundle level. The impact can often be eliminated by deploying compensating changes in other components, leading to an invariant bundle-semantics, where the consumers are not affected by internal modifications [Orman 2008a; Vigano and Colombetti 2007].

Consider an airline as a collection of services. An international flight delay can potentially cause large scale disruption in operations, including the connecting domestic flights, crew assignment schedules, gate utilization, ground transportation, customs requirements, and food and lodging services. A bundle semantics system can adjust all the components of a bundle to ensure that the bundle, as a whole, remains consistent without disrupting the organizational goals. It accomplishes this by replacing some components with semantically equivalent, or approximately equivalent, components that can deliver the same organizational goal. If the consistency cannot be maintained, the semantic system can calculate the impact of any structural change on the overall semantic goal of the bundle and present tradeoffs. For example, delaying some domestic flights or replacing some crews with others possessing the same skills can resolve the problem without affecting the overall goal. Otherwise, the manager may be presented with some tradeoffs about the impact of service modifications and substitutions on the bundles that utilize those services. If enough detail is known about the organizational preferences, alternative bundles can be selected and implemented optimally [Hyvönen et al 2004; Orman 2008a].

- d. Organizational components can be shared by multiple organizations in their ongoing operations, facilitating reliable and dynamic outsourcing of complex services. Consider a drug testing service in a pharmaceutical company. The goal of the service is to determine the effectiveness of a drug through laboratory, animal, and human testing. There are multiple methods to achieve this goal with varying levels of risk, efficiency, and cost. Typically, organizations make a top-level decision about the method to use and the procedures to follow, and enforce that decision organization wide. A flexible organization would delegate that decision to multiple sub-organizations that may accomplish this goal in different ways, and compete among themselves for resources. Each sub-organization in turn may delegate various components of the task to lower level, competing, sub-sub-organizations. Each of these sub-organizations may belong to a different business that specializes in a particular method of testing. Each sub-organization can also serve many organizations which similarly delegate their testing needs, and it competes for resources with many other sub-organizations that provide semantically similar services, by using different methods and structures to achieve the same goal [Mowshowitz 2003].

- e. Formal descriptions can be used as binding contracts by: forcing the lower level sub-organizations to inherit the contracts of the higher level components, enabling the higher level units to enforce constraints on their subcomponents, and by holding higher level components liable for the breach of contracts of their sub-organizations, to establish wide ranging inter-organizational trust.

Consider a high level, formal relationship between a collection of insurance companies and an association of hospitals. Such a relationship can be used to support ad hoc transactions between individual patients and doctors through their membership in affiliated insurance programs and accredited hospitals. Such ad hoc transactions between patients and doctors can be conducted with confidence and trust because of their memberships in higher level organizations that enforce formal service descriptions, provide guarantees, inspect quality, and intervene to resolve disputes. Such a high level contract would need to be more than a mere reputation of the participants. It would require a formal description of the contract, its valid audience, the goals it supports, and its structural components. Such high level contracts can only be built and maintained efficiently if they can be composed from lower level components and share those components with other organizations. The failure of HMO's as a health care delivery service, for example, can be traced to their inability to build a multilevel and flexible system, with formally defined and replaceable components that can be aggregated efficiently to build large and formally defined, but flexible and customized bundles.

These flexible organizations are hierarchies, but they operate like mini-markets at each level of the hierarchy. They can also be viewed as markets, but a hierarchical structure is imposed on them in which the higher level market transactions restrict lower levels in terms of their scope and goals. The resulting structures can also be viewed as an extensive outsourcing of services at multiple levels while a tight coordination is implemented through formal semantics. The parallel sub-organizations may or may not be parts of the same organization. In fact, such structures blur the boundaries of an organization by creating a collection of semi-independent sub-organizations at multiple levels that cooperate at their goal level, but compete within the constraints of that goal. This kind of organizational fluidity can be achieved either within a single organization through structural design, or within a marketplace through long lasting transactions and contracts [Rust 2004; Sawhney et al. 2005].

IV. SERVICE SEMANTICS

Having created an ontology for service classes, we now turn to the semantics of individual services. Semantics has received a great deal of attention in the context of the Semantic Web. Description of web documents has been critical to the development of WWW, and more effective description through semantic models is an ongoing concern to facilitate more effective search and utilization of the web resources [Kashyap et al, 2008]. Extensible Markup Language XML was a critical development. It created a tagging system for document components, attaching keywords to document components, which in turn allowed querying for document components on the basis of those tags. The tags can be viewed as attribute names attached to values where the values are document components, leading to a collection of attribute-value pairs. For example the text "Ford Focus" can be tagged with the keyword "car model", leading to an attribute-value pair of "car model-Ford Focus". Such attribute-value pairs have been the basis of many semantic models ranging from semantic networks to semantic data models [Gomez-Perez et al, 2004].

Attribute-value pairs of XML can be arranged hierarchically to represent the hierarchical arrangement of document components [Kashyap et al. 2008]. For example, a car document and its Ford subcomponent are in a hierarchical relationship. These document components are stored as XML databases and queried using XML query languages such as XQuery and XPath that retrieve document components when related components in an XML hierarchy satisfy some given attribute-value conditions. For example, the fuel economy mpg of a Ford Focus hatchback can be retrieved from an XML document about cars with the tag sequence of car. ford. focus. hatchback.mpg=?

by using a stylized language similar to XPath where every node on the path is a node on a type hierarchy or an attribute name. A number of languages such as XPath are available to query XML databases, and similar attribute-value based query languages have also been very successful with relational and object-oriented databases. XML databases are supported by Data Type Descriptions (DTD) which act as schemata to ensure consistent use of tags. DTDs can also be extended to universal ontologies that define terms, place them in type hierarchies, and attach constraints to limit the acceptable values [Gomez-Perez et al. 2004].

XML provides an infrastructure on top of which more elaborate semantic structures can be built. XML's simple attribute-value pairs proved inadequate for universal identification and description of web resources, as well as reasoning and inference with such widely distributed objects. Resource Description Framework (RDF) was built on top of XML. It extended XML with Universal Resource Identifiers (URI) for unique identification of all resources, and with entity-attribute-value triplets where entities are uniquely identified by URIs and the values can be other entities [Kashyap et al. 2008]. Such a recursive definition increases the semantic power of the model and the complexity of

the resources it can describe. Finally, the resource descriptions have to be placed in central repositories to facilitate search, selection and retrieval. This capability requires searchable resource description databases. Many of these were developed, such as the Universal Description Discovery and Integration (UDDI) framework [Erl 2005]. They all rely on XML for their database structure, and XML languages for search, selection, and retrieval.

This article follows in the tradition of semantic resource description, but focuses on a specific resource: services. Consider a travel agency. The travel services are described by the provider, and the record includes a description of the trips, their cost, the payment method, and the buyer's identity. The buyer may also record the service in his personal records for tax and record keeping purposes including the price, the payment method, and the seller's identity. Third parties, such as insurance companies and government agencies, may also record the same data for insurance and security purposes. Such transaction databases of large businesses and government agencies grow quickly to terabyte sizes and become difficult to use. But despite their formidable size, they fail to capture some important components of the service semantics, which contributes to their usability problems. For example, they fail to capture and record why the trip was taken and for what purpose, i.e., its goal semantics. They fail to capture and record whether the trip met the expectations, and which objectives were satisfied and to what extent, i.e., its outcome semantics. They fail to record and capture secondary services such as insurance, hotels, restaurants, and shopping services and the relationships between those secondary services and the original service, i.e., its bundle semantics. They fail to capture and record what other trips were considered, but rejected by this consumer in lieu of this service, and why they were rejected, i.e., its alternatives semantics. Finally, they fail to capture and record the conditions and the events that triggered the trip, i.e., its time semantics.

A typical service is described by what resources are exchanged, and who the participants are. These can be called "what" and "who" semantics. A complete semantic description would have to elaborate on what and who semantics by describing the resources and the participants in detail. It would also have to add many more components, such as the goal of the service, called the "goal" or "why" semantics; the use of the resources involved and the resulting satisfaction level, called the "outcome" or "use" semantics; the location of the service, called the "location" or "where" semantics; the payment mechanism, called the "payment" semantics; secondary services supporting the original service, called the "bundle" or "what else" semantics; and the alternative services that have been considered and rejected, called the "alternatives" semantics. All of these components can be implemented as the attributes of an RDF resource, where the resource is a service, instantiated from the service ontologies of Section 3. The values attached to these attributes can be quite complex, and require the full recursive power of RDF, where the attribute values can themselves be entities with their own attributes, and with complex constraints and relationships imposed on them. Such complexity is rare in corporate data stores, or even in web documents, and it motivates the complex model proposed in this article. The complexity is exacerbated by the difficulty of collecting, structuring, searching, and utilizing such complex and detailed information about services.

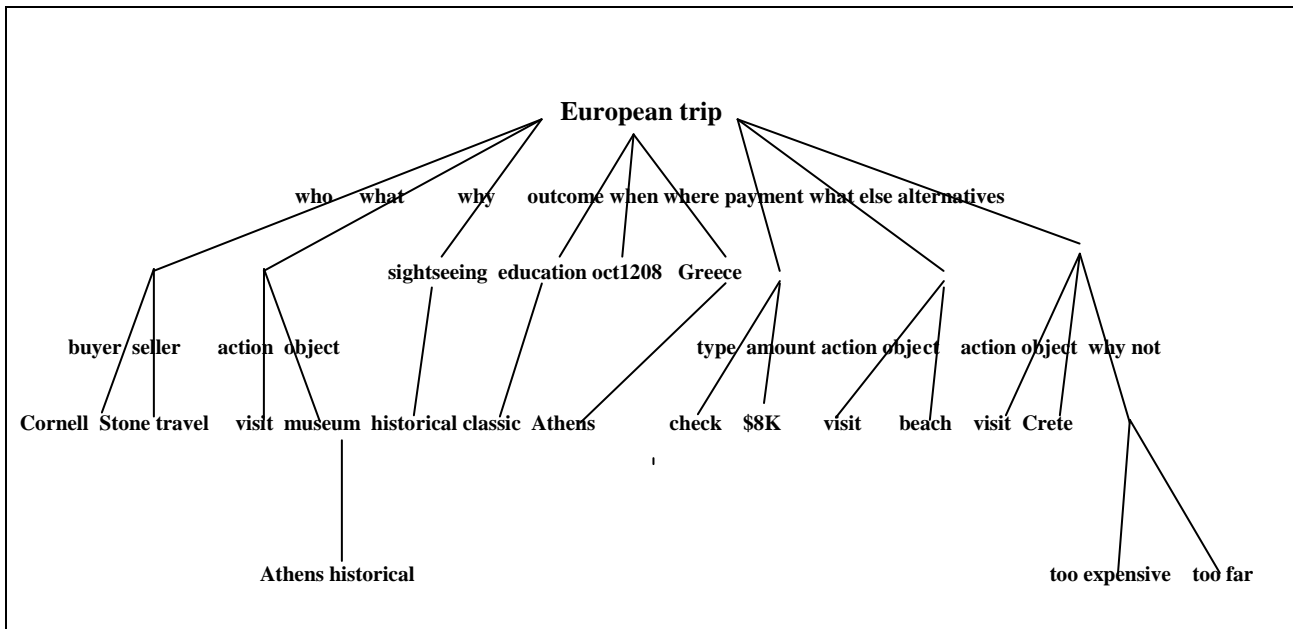


Figure 2. The Model of a European Trip Where the Labeled Edges are Attributes and the Unlabelled Edges are Subtypes

The basic model of Figure 4.1 is a simple set of entity-attribute-value triplets, such as service-why-sightseeing, that can be implemented as a relational database with commercial software that stores and maintains the data and facilitates retrieval with its query language [Sheth et al. 2006]. But there are a number of complications. The first complication is the need to disambiguate the terms used for universal consistency in a large scale distributed system. The solution for entities is to use Universal Resource Identifiers (URI) as suggested by Resource Description Framework (RDF) [Erl 2005]. Figure 4.1 contains URIs at all unnamed nodes and at the service node. The solution for attribute-value pairs is to use ontologies to disambiguate them, such as Web Ontology Language (OWL) attached to RDF. In effect, all terminal values in Figure 4.1 point to a node in the vacation ontology of Figure 3.1, and attach a dictionary definition and constraints to it [Hearst 2006].

This structure is sufficient to achieve the goals set out in Section 1. The objective of locating relevant components of a service is achieved by what and what else semantics, since they identify components. The objective of communicating goals and plans is achieved by why semantics as it identifies goals unambiguously. The objective of combining and coordinating components is achieved by what else and alternatives semantics since they identify complements and substitutes of service components. Finally, the objective of establishing trust is achieved by the outcome semantics as it captures and records the results achieved by others with similar goals. These features can be implemented with a relational database for the attribute-value pairs, OWL ontologies for defining attribute values unambiguously in dictionaries and type hierarchies, and special pointers inserted into the database linking database values to the ontology entries, as shown in Figure 4.2.

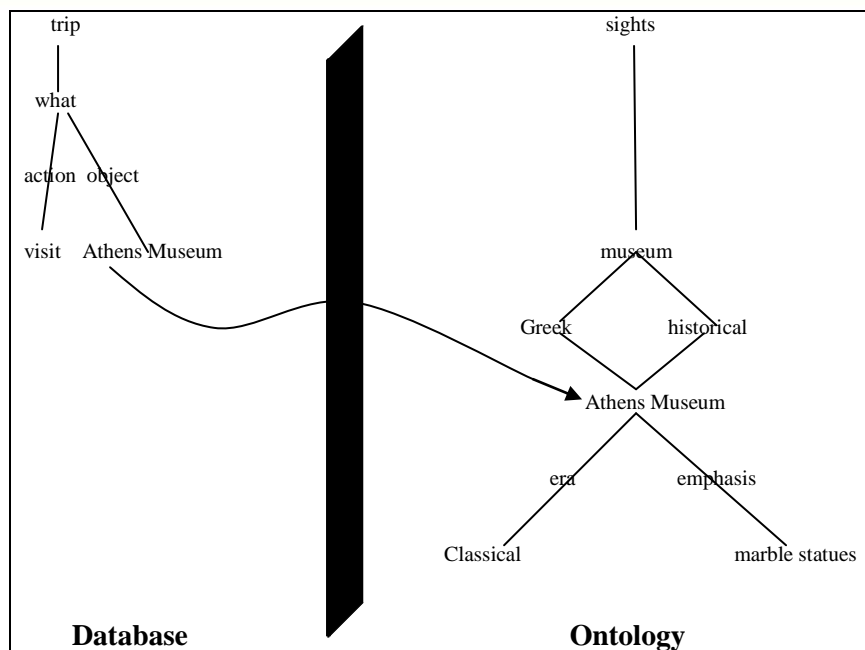


Figure 3. Extending the Relational Database with Pointers to Ontologies to Disambiguate Terms and to Describe Them.

The second complication is the existence of ad hoc and multi-valued attributes. For example: some museums may have awards and reviews as attributes, others may have openings and exhibits, some may not have any, and some may have them as multi-valued. Some attributes may be very sparse, or even unique to individual museums, such as the cost of a new Zeus exhibit. These ad hoc attributes can be implemented as RDF entity-attribute-value triplets, as relational databases would have great difficulty accommodating them. An additional difficulty here is the need to link the relational and the RDF components, and to search them simultaneously to provide complete responses to questions. This task can be accomplished by creating a high level RDF model that encompasses both the relational and the RDF components. The queries are posed to this RDF model first, and then translated into queries against the components for more efficient processing. This is a classic, distributed and heterogeneous data store example which characterizes most web services, and the research in this area is very active [Erl 2005].

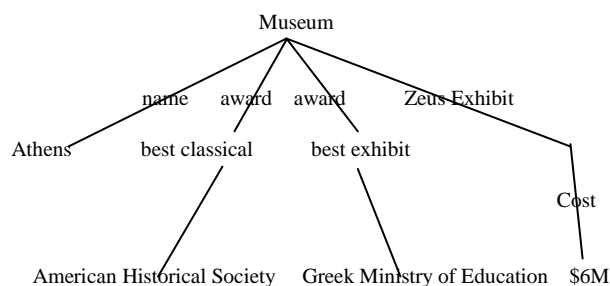


Figure 4. The Model of a Museum Containing Ad Hoc, Multi-valued, Sparse, and Unique Attributes

Finally, the third complication is the attribute values being services themselves, leading to a recursive model. Both bundle semantics and alternatives semantics are implemented as attributes of services which are services themselves, and they point back to the service database to identify related services. Alternatives semantics consists of virtual services, since alternative services are services that have never taken place, but merely considered. These alternative services also have "why not" semantics that explain the reasons for the rejection of those services. These

recursive relationships with ad hoc and sparse attributes can be implemented as RDF triplets, and stored in RDF data stores [Hearst 2006].

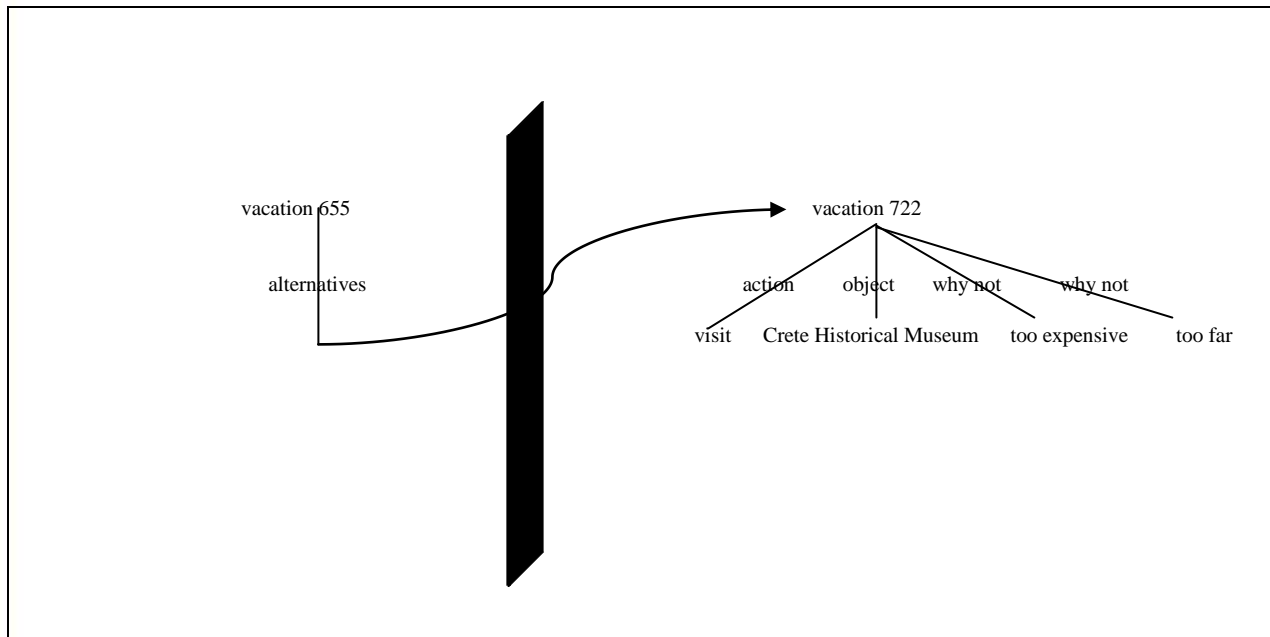


Figure 5. The Model of Alternative Services Considered by a Consumer in Lieu of Visiting Athens Historical Museum.

The complete semantic model then, is an RDF schema whose implementation contains relational and RDF components and ontologies with linkages among all three. In effect, the implementation involves a relational database extended with an RDF data store to accommodate ad hoc, sparse, and recursive attributes. It also involves pointers embedded into the relational database and the RDF data store, pointing to OWL ontologies or recursively to other relational and RDF components. The pointers to OWL ontologies are implemented as XML path expressions that point to specific locations in the XML type hierarchies of the ontologies, which identify attribute values unambiguously, and attach constraints and dictionary definitions to them. The pointers to other relational records or to RDF triplets contain merely the URIs of those entities since all entities, whether they are relational or RDF, are identified uniquely by their URI.

Efficient implementation of such an extensive semantic database is a challenge, especially because its large scale exacerbates its structural complexity. It could be wholly implemented as an RDF data store of entity-attribute-value triplets and associated OWL ontologies, but RDF data stores do not scale up efficiently. Implementation as a combination of relational databases and RDF data provides a more efficient solution by separating the highly structured transaction data with pre-specified attributes from ad hoc data with unique attributes and sparse content. Highly structured standard services can rely on relational databases and their efficient indexing structures. The only source of inefficiency in that scenario is the data values that are XML path expressions pointing to OWL ontologies.

Entering such complex values into a relational database, indexing, and querying them efficiently are significant challenges. Such an expression can be entered directly into a database, or it can be entered as a pointer to the right location in the ontology. A value in the service database can then be an XML path such as “outcome.education.historical”, pointing to a location in an OWL data type hierarchy about education. Entering a pointer would have to be supported by a user interface that allows navigating type hierarchies of ontologies.

Entering path expressions would require resolving ambiguities when only a partial path expression is entered, or superfluous nodes are specified. For example, the complete path above may be “service.outcome.education.historical”, or “education.historical”, depending on how the ontologies are organized, and the correct expression must be automatically derived from the partial expression entered by the user.

Alternatively, the user may have to be guided in entering such path expressions, and whether they contain new values, or point to existing ontologies. Path expressions may have to be entered one node at a time. At each step, the users are presented with a set of possibilities from the existing ontologies, and are given an opportunity to define new values that do not violate the type restrictions. For example, the outcome attribute of a museum visit has a number of possible values, one of which is “education”. The users need to be presented with a good collection of

possible values when they arrive at the outcome node. For a different service involving shopping, “education” may not be an existing relevant attribute. The outcome for that type of service may involve “historical gifts”, which may involve an education attribute. The options presented to the user, and the sequence of those options are specific to each transaction type, and the choices at each step limit the choices available in subsequent steps. Such navigation of ontologies is critical to populating the database with correct path expressions and pointers.

V. VIRTUAL COMMUNITIES

Communities are groups of people with common characteristics working toward a common economic, social, or political objective. They are special organizations characterized by social interaction, political organization, or consumption rather than production of goods and services. Virtual communities are flexible communities dynamically built from the components of existing communities. They are characterized by quick assembly and disassembly, but also by sharing of components. Consider a community of information technology professionals. It can be built from the components of various engineering and scientific communities, while sharing members with a variety of communities such as engineers, scientists, social scientists, and mathematicians [Jeffcote 2003].

Communities can vary in degrees of virtuality. At one extreme are families that exhibit no virtuality since they are long term, static, and very slow changing (except for births, deaths, and marriages). At the other extreme are group-shopping Web sites on which complete strangers can form a temporary community for the purpose of conducting a single transaction, then disassemble and possibly join other communities. The degree of virtuality in a community is determined by the size and complexity of the shared components. Sharing is critical because building each community from scratch, and managing and maintaining them separately would be prohibitively expensive in large scale. This is why electronic communities proposed nearly a decade ago have never been very successful. This is also why information technologies and design architectures have a crucial role to play.

Communities can be built from existing components, greatly reducing the cost of building communities. They can share components with other communities, greatly reducing the cost of managing and maintaining them. A second reason why virtual communities have not proliferated during the past decade is because, although they are highly responsive to changes in the environment, they have difficulty following long term plans and strategies. Formal description of communities with multidimensional ontologies is likely to remedy that problem by increasing the size and complexity of shareable components, and by monitoring and constraining the lower level components by the higher level components of the hierarchical type structure, as discussed in section 2 [Jeffcote 2003; Mowshowitz 2003].

Communities consume services, and can be described in terms of the services desired. For example, an educational community is interested in lectures, courses, concerts, and art exhibits. The aggregation of these services defines the community. The community’s objective is to acquire these services and deliver them to its members. The community itself can be self governing, as in cooperatives; or it could be a business endeavor where professional managers serve the interests of the community in an effort to make a profit. In either case, communities can be incorporated as legal entities, and they can hold assets and liabilities separate from their members. Members can share in the ownership of assets equally, as in cooperatives. Alternatively, they can buy shares in the community as in mutual funds [Orman 2008].

Communities can engage in transactions on behalf of their members. Such aggregate transactions can be a source of efficiency since a single transaction on behalf of all members reduces transaction costs. The search for the transaction can be done professionally by the community manager for the whole community, and the transaction can be executed on behalf of all members with a single contract. Communities can also increase the bargaining power of consumers. The traditional interaction between consumers and businesses is never a relationship of equals. Consumers are always at a disadvantage, both in terms of their knowledge of the products and services (information asymmetry), and in terms of their control of the market and bargaining power (power asymmetry). Communities equalize that imbalance.

Finally, communities can protect the privacy of their members. A community engaging in transactions on behalf of its members isolates their identity and characteristics from the marketplace. For example, mutual funds and investment communities protect the identities of their members from the marketplace by isolating them from the transactions that the fund or the community executes on their behalf [Jeffcote 2003; Rust 2004].

Communities often share not only transactions, but also the resources acquired by those transactions. By sharing resources they provide an intermediate approach to resource ownership between private and state ownership. A sailing community that shares a sail boat, or a racing community that shares a race car can lead to a very efficient utilization of a capital asset without the large capital requirements of individual ownership, and without the regulatory

requirements of state ownership. In that respect, they are similar to rental businesses that acquire resources for many consumers to share. However, they accomplish the same objective without the high transaction costs of rental businesses by replacing short term individual transactions with long term aggregate contracts. Such an intermediate solution between private and public ownership would be most appropriate with moderately high value capital assets and consumer goods, wherein individual ownership is too burdensome, yet state ownership is too formal and bureaucratic [Lechner 2002] .

The ability to share transactions and resources has important consequences. Businesses and consumers can now engage in multilevel aggregate transactions where high level transactions constrain low level transactions, and low level transactions inherit the properties of high level transactions. This leads to a rich tapestry of transactions with increased efficiency of search and execution. Vendors involved in such aggregate transactions are a collection of businesses that commit themselves collectively to the aggregate transaction, but then the components are supplied by individual businesses according to a pre-established protocol. A typical example is an AAA automobile association service agreement which is fulfilled by the closest service station to the consumer requesting the service at any given time. Extending such transactions to a variety of maintenance services ranging from home repair to electronics should be straightforward. Extending them to consumer goods, transportation and travel services requires a careful and formal description of services to match the offers with the needs. An example in travel industry is the time share of vacation rentals, which are currently quite limited in sharing of both transactions and the properties.

Extending this model to sharing among many businesses and individual homeowners would require detailed and formal descriptions of the services offered and demanded. Such detailed, formal descriptions cannot be efficiently produced and monitored without an extensive infrastructure that allows formally described components to be assembled into meaningful aggregates, and disassembled into individual transactions with great speed and efficiency. Such an infrastructure requires capturing complex semantics at multiple dimensions and multiple levels of aggregation, and also an interface to present it in an intelligible fashion to potential users.

A typical example of such an extension would be a university engaging in a transaction to provide spring break vacations for its entire faculty in the Caribbean for the next five years. Each vacation would involve various time-share arrangements to accommodate the individual requirements, desires, and the family size, but the aggregate transaction would have a separate semantics sufficient to execute the transaction without the details of individual requirements. Many intermediate groupings would be useful in segmenting the large population into homogeneous groups and developing intermediate semantics, such as, those who can get away for only 3 days versus 7 days, or those who have children versus those who are singles or couples. A multi-level and multi-dimensional ontology would be necessary to capture all groupings and all relevant attributes at all levels, yet have an abstract semantics for the aggregate [Jeffcote 2003; Sawhney 2005].

One approach to present such semantics is through virtual individuals. A virtual individual is a fictitious individual that personifies the community. It is built from components of real, individual descriptions. It has a legal identity, assets, liabilities, rights, and responsibilities. On one hand, it is similar to a political candidate who personifies a community and represents its interests. On the other hand, it is similar to a corporation because of its legal identity separate from the individuals it represents.

The virtual individual is implemented by the community it represents, and can be a cooperative of real individuals or a separate business endeavor (as discussed earlier). The real individuals comprising the virtual identity benefit from the transactions of the virtual individual to the extent that they participate in the virtual identity. Their benefits are limited by their contribution, and their liabilities are limited by their commitments in a manner similar to other legal entities representing members, such as corporations.

A typical example of a virtual individual representing a community is a temporary employment agency that finds employees for businesses. A business hires the temp agency to fill a position for a period of time. The position is formally described in the persona of an ideal individual. The temp agency fills the position with a virtual individual, which could be a different real individual every day. The group of real individuals collectively filling the position form a community represented by the virtual individual.

Many service businesses operate similarly. When one hires a plumbing service to provide home maintenance services, one interacts with a virtual plumber. Different physical plumbers show up for different appointments, and many plumbers participate in the persona of one virtual plumber.

These examples relate to traditional businesses using a community of employees represented by a virtual individual. To extend this model to consumer communities would require a complex infrastructure and a multidimensional

semantic system. Consider a community of fishermen interacting with a group of boating businesses in a single transaction, for a variety of boating needs, by a variety of fishermen, in various fishing trips. A hierarchical structure of various types of fishermen from various fishing communities, types of trips designed by various adventure communities, and types of boats offered by various boating businesses, all interacting in a single aggregate transaction to accommodate a large variety of services would require a complex, multi-dimensional ontology and a complex transaction. The advantage would be considerable efficiency in search, transaction costs, privacy and trust [Hyvönen 2004].

Employment of virtual individuals can create flexible work schedules and novel outsourcing opportunities. A group of real individuals supporting the employment of a single virtual employee can provide a reliable employee to the employer and a flexible work schedule for the employees. The virtual employee can be viewed as outsourcing a job to a community, since a community of employees provides labor viewed as a single employee by the employer. A virtual corporation can be formed entirely of virtual employees, where each virtual employee is supported by a community of real employees, and various collections of virtual employees themselves can be viewed as higher-level virtual employees. At the extreme case, a virtual corporation itself can be viewed as a virtual employee, built from other virtual employees, and the distinction between businesses and employees begins to disappear. Such flexible work environments are information intensive and require formal descriptions of both the jobs and their skill requirements, as well as the employees, and their qualifications and needs. Dynamic formation of such communities requires multi-level and multi-dimensional ontologies to form larger communities from existing components because:

- Formal description of tasks is required so that all community members can dynamically fill the position and perform consistently as substitutes for each other, alone or in groups.
- Formal description of employees is required to guarantee that all members are qualified for the job they are assigned. The assignment can be done dynamically and automatically.
- A coordination and enforcement mechanism is required that allows community members to allocate and schedule work, and to share compensation, all dynamically and without extensive planning, preparation, and lead time.

VI. VIRTUAL TRANSACTIONS

Virtual transactions are high level transactions built by aggregating lower level transactions dynamically and at multiple levels. Corporations engage in virtual transactions routinely. A year end dividend payout or a corporate merger are examples of high level transactions that are aggregations of many individual dividend payouts, or many shareholder stock-swap transactions, respectively. Consumers, on the other hand, are not always able to engage in virtual transactions. This greatly limits their economic efficiency and power.

Transactions can be aggregated over three separate dimensions: goods and services, time, or consumers. Bundling an automobile insurance transaction with a home insurance transaction is an aggregation of services. Bundling home insurance policies over many consumers, such as a bulk insurance policy for all employees of a university, is an example of aggregation over consumers. Automatically updating and renewing an insurance policy every year is an example of aggregation over time.

One can aggregate over multiple dimensions, at multiple levels, and sometimes conditionally, leading to complex multidimensional hierarchies. Consider the sale of a 10- year, home/automobile insurance policy to all employees of a university who have not had a traffic accident during the past five years, whose home coverage is 80 percent or 90 percent of their home value depending on the neighborhood, and is renewed every year at the level of inflation. A further aggregation of such transactions to all insurance needs of all academic employees, over their lifetime can be accomplished through multiple levels of conditional aggregation of various transaction types. Such high level transactions can be described as abstract lifestyle choices, and then automatically translated into specific transactions for specific consumers [Orman 2008].

Such high level aggregate transactions can be fulfilled by multiple vendors over time. Various intermediaries commit to such high level transactions at various levels of abstraction. They then arrange for lower level intermediaries and vendors to fulfill individual transactions. Some vendors may not even exist at the time of the commitment, but are recruited at a later time to fulfill the obligations. The specific products and services may not even exist at the time of the commitment, and such abstract commitments by a large number of consumers to future products and services may encourage experimentation with future business models.

Formal descriptions further encourage experimentation to judge the potential of new products and services. Surveys and polls are often used for such experimentation, but without formal descriptions such experiments often fail, especially if the new products and services are dramatically different from the familiar ones. Similarly, virtual spaces like Second Life are suggested for large scale experimentation. However, three-dimensional visualization is not sufficient to describe most products, services, organizations, or consumer needs. Effective formal descriptions are likely to improve product development and market research considerably by providing a platform to test consumer interest reliably, long before a product or service is developed. They are also likely to be useful in testing a large variety of alternatives to existing products and services by providing simulations of them in parallel with real products and services. This allows consumers to compare and contrast the real products with the simulated ones, as they use the real products. It is also possible to extend such simulations to organizations and communities, and to experiment with alternative organizational forms, while the real organization is operating and the simulated organizations run in parallel [Malone et al. 1999].

Virtual transactions blur the distinction between shareholders and customers by allowing large scale experimentation by consumers with new products and services. Long term aggregate transactions are commitments by large groups of consumers to future products and services, and as such, they are similar to investments. The ability of large communities to support future products and services long before they are developed, on the basis of high-level goals and targets, allows experimentation by suppliers within the constraints of those goals and targets. Consider a long term subscription to home security services by a large community of homeowners. Such long term commitments by large communities are large and long term investments. They provide the necessary funds to virtual vendors to continually improve their services by replacing some organizational components with new vendors, and by experimenting with new technologies and business models, without the need to resell, re-market, or seek venture capital for every change [Sawhney et al. 2005].

VII. VIRTUAL INSTITUTIONS

Social and political institutions such as governments, churches, schools, and families are long lasting and slow changing organizations. They often demand exclusive membership, form highly bureaucratic and political power structures with vested interests, and are slow to adapt to the changing environment. Their members cannot easily leave these organizations without incurring a high cost. However, they cannot easily effect change in these organizations either, due to built in inertia and vested interests, as well as the risks associated with changing such organizations that are critical to social stability. Consequently, these institutions slowly, but steadily become obsolete and stop serving the interests of their members at large. Any major change in these institutions is slow, contentious, and highly political, with long periods of debate and controversy before the change takes place [Hammer and Champy 2004; Smith and Finger 2003].

Modern social institutions are critically important to the well being of their members, yet they have great difficulty experimenting with new models. Incremental changes are often short term oriented and lack long term vision. Large scale change is rare since it displaces powerful interest groups, creates instability, and threatens the well being of many members.

In primitive societies, these institutions were also critically important. However, they were not very large and powerful, and fierce competition among societies quickly eliminated those with dysfunctional institutions. In modern societies, these institutions and the societies that support them have become very large and entrenched, and the competition among them is very limited. Most people simply do not have the option, the opportunity, or sufficient knowledge about the alternatives, to change their nationality, system of government, religion, or their family [Garcia-Camino 2006].

Virtual institutions with formal descriptions and hierarchical type structures can create flexible societies where experimentation is encouraged, and people can move between institutions freely without major disruptions in their lives. Such a flexible social environment requires a number of fundamental components:

- a. Formal descriptions of institutions in terms of the services they provide. An institution is usually a large collection of services which defines both the institution and the benefits of membership. For example, a church is a collection of well defined, spiritual, social, and community services. It is organized to provide those services, and its members are entitled to those services due to their membership. Formal descriptions of such services allow members to compare and contrast those provided by alternative institutions, rather than being locked into one institution. Formal descriptions facilitate efficient search and comparison [Garcia-Camino et al., 2006; Hyvönen et al. 2004].

- b. A hierarchical-type structure and an ontology to identify and describe service components, compare and contrast them, build larger components from the smaller ones, and share components within and among institutions. Social services provided by various churches, synagogues, and mosques are often quite similar each other. They could be separated from the spiritual services, formally described, and their minor differences formally identified. This would allow multiple religious institutions to quickly join forces when necessary, share institutional components, and provide comprehensive social services when their goals are aligned. The same type of dynamic alignment can also be useful in political action and community building [Vigano and Colombetti 2007].
- c. A mechanism to allow members to move freely between institutional components while maintaining their membership in higher level components. Such a mechanism would allow members to mix and match components to build customized institutions, or to experiment with new components incrementally while maintaining their participation in other components of the institution.

This capability would provide an effective outlet for organizational dissent, which would be a compromise between the two commonly available options: voice and exit [Hirschman 1970]. The members would not have to leave the organization completely to express dissent; nor would they have to engage in political organization and protracted debate to sway the organization in their direction. They could simply vote with their feet, and leave those components that they find unsatisfactory. The further ability to take some resources with them would impact the organization, creating an automated mechanism for dissent, experimentation, and institutional evolution, all of which are sorely missing in modern institutions.

For example, a church member can participate in social services of one church while attending the spiritual services of another. Citizens can leave the social services of one government for another, yet continue to support the military services of that government. Services of various governments themselves can be combined to build larger, international services dynamically. Social and military services of two governments can be combined to provide retirement benefits in one country to the veterans of another country. Families can separate child care and education services from the financial and emotional services, and then combine and share child care services with other families in an extended network of families, as the need arises due to emergencies, family crises, geographical moves, professional needs, or vacations. Such flexibility in institutional design requires formal description of services in multi-dimensional ontologies, and a mechanism design to combine institutional components dynamically to build larger components [Garcia-Camino et al. 2006; Hirschman 1970].

VIII. CONCLUSIONS

Successful implementation of virtual organizations requires a complex combination of technology infrastructure and organizational design. The approach proposed in this article is based on service semantics. Organizational components can be viewed as bundles of services, and their semantics can be captured in multi-dimensional ontologies. The type hierarchies of these ontologies can be used to formally describe organizational components, and to assemble organizational components into larger units while maintaining a well defined semantics. The infrastructure necessary for a large scale implementation of this architecture is complex, and development is in the early stages. There are many commercial tools for ontology development, but using these tools to develop large repositories of components is in its infancy. This architecture requires large commercial repositories of organizational components that can be searched and discovered, and readily utilized to build new components. Development of such repositories is critical to large scale implementation, and they require not only easy to use semantic tools, but also scalable repositories to store component descriptions in an easily searchable format. Both of these requirements need extensive development, experimentation, and design of new interfaces in addition to the design architecture proposed in this article.

REFERENCES

- Alonso G. (2003). *Web Services*, Springer Publishing.
- Camarintha-Matos L. M., H. Afsarmanesh, and M. Ollis (2005). *Virtual Organizations: Systems and Practices* Springer-Verlag Pub.
- Chesbrough H. and J. Spohrer (2006). "A Research Manifesto for Services Science," *Communications of the ACM* (49)7, pp.35-40.
- Cherbakov L. et al. (2005). "Impact of Service Orientation at the Business Level," *IBM Systems Journal* 44(4), pp. 653-668.



- Devedzic V. (2002). "Understanding Ontological Engineering," *Communications of the ACM* 45(4), pp. 136-144.
- Elfatraty A. (2007). "Dealing with Change: Components versus Services," *Communications of the ACM* 50(8), pp. 35-39.
- Erl T. (2005). *Service Oriented Architecture (SOA): Concepts, Technology, and Architecture* Prentice Hall.
- Franke U. J. (2002). *Managing Virtual Web Organizations in the 21st Century: Issues and Challenge*. IGI Publishing.
- Garcia-Camino A. et al. (2006). "Electronic Institutions," *ACM SIGECOM Exchanges* 5(5), pp. 33-40.
- Gomez-Perez A., M. Fernandez-Lopez, and O. Corcho (2004). *Ontological Engineering*. Springer-Verlag.
- Hammer M. and J. Champy (2004). *Reengineering the Corporation: A Manifesto for Business Revolution* Collins Pub.
- Hearst M. A. (2006). "Clustering versus Faceted Categories for Information Exploration," *Communications of the ACM* 49(4), pp. 59-61.
- Hirschman A. O. (1970). *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States* Harvard University Press.
- Hyvönen E., S. Saarela, and K. Viljanen (2004). "Application of Ontology Techniques to View-Based Semantic Search and Browsing," *Proceedings of ESWS European Semantic Web Symposium* Springer Verlag.
- Jeffcote R. (2003). "Technology and Utopia," *Journal of Interdisciplinary and Cross-Cultural Studies* 3.
- Kashyap V., C. Bussler, and M. Moran (2008). *The Semantic Web: Semantics for Data and Services on the Web* Springer Publishing.
- Lechner U. and J. Hummel (2002). "Business Models and System Architectures of Communities," *International Journal of Electronic Commerce* 6(3), pp. 41-82.
- Malhotra Y. (2000). *Knowledge Management and Virtual Organization*. Idea Group Pub.
- Malone T. et al. (1999). "Tools for Inventing Organizations: Towards a Handbook of Organizational Processes," *Management Science* 45(3), pp. 425-443.
- Mowshowitz A. (2003). "Virtual Organization: Towards a Theory of Societal Transformation Stimulated by Information Technology," *Communications of the ACM* 4(11), p. 2-4.
- Orman L. V. (2008). "Service Semantics, Structure, and Design," *E-Service Journal* 6(2).
- Orman L. V. (2008a). "Browsing Databases with Constraint Hierarchies," *Electronic Commerce Research* 8(1), pp. 29-56.
- Ren M. and K. J. Lyytinen (2008). "Building Enterprise Architecture Agility and Sustenance with SOA," *The Communications of the Association for Information Systems* 22(4).
- Riki A., Y. Morikawa, and L. Kerschberg (2004). "MAKO: Multi-Ontology Analytical Knowledge Organization Based on Topic Maps," *Proceedings of DEXA Conference* pp. 459-463.
- Rust R. K. (2004). "A Call for a Wider Range of Services Research," *Journal of Service Research* 6(3), pp 2-11.
- Sawhney M., G. Verona, and E. Prandelli (2005). "Collaborating to Create: The Internet as a Platform for Customer Engagement in Product Innovation," *Journal of Interactive Marketing* 19(4), pp. 4-17.
- Sheth A., K. Verma, and K. Gomadam (2006). "Services Science: Semantics to Energize the Full Services Spectrum," *Communications of ACM* 49(7), pp. 55-61.

Smith H. and. P. Fingar (2003) *Business Process Management: The Third Wave* Meghan-Kiffer Press.

Vigano F. and M. Colombetti (2007). "Symbolic Model Checking of Institutions," *Proceedings of ICEC Conference on Electronic Commerce*.

Woods D. and T. Mattern (2006). *Enterprise Service Oriented Architecture* O'Reilly Publishing.

ABOUT THE AUTHOR

Levent V. Orman is a professor of Information Systems at Cornell University. He received a Ph.D. in Information Systems from Northwestern University. He has published extensively on electronic commerce, database management, and decision support systems. His articles appeared in a variety of journals ranging from *Information Systems*, *Acta Informatica*, and *IEEE Transactions on Knowledge and Data Engineering*, to *MIS Quarterly*, *Journal of MIS*, and *Electronic Commerce Research*. He serves on the editorial board of the *Journal of Information Technology and Management*.

Copyright © 2009 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712, Attn: Reprints; or via e-mail from ais@aisnet.org.



Communications of the Association for Information Systems

ISSN: 1529-3181

EDITOR-IN-CHIEF

Ilze Zigurs
University of Nebraska at Omaha

AIS SENIOR EDITORIAL BOARD

Guy Fitzgerald Vice President Publications Brunel University	Ilze Zigurs Editor, CAIS University of Nebraska at Omaha	Kalle Lyytinen Editor, JAIS Case Western Reserve University
Edward A. Stohr Editor-at-Large Stevens Institute of Technology	Blake Ives Editor, Electronic Publications University of Houston	Paul Gray Founding Editor, CAIS Claremont Graduate University

CAIS ADVISORY BOARD

Gordon Davis University of Minnesota	Ken Kraemer University of California at Irvine	M. Lynne Markus Bentley College	Richard Mason Southern Methodist University
Jay Nunamaker University of Arizona	Henk Sol University of Groningen	Ralph Sprague University of Hawaii	Hugh J. Watson University of Georgia

CAIS SENIOR EDITORS

Steve Alter University of San Francisco	Jane Fedorowicz Bentley College	Jerry Luftman Stevens Institute of Technology
--------------------------------------------	------------------------------------	--------------------------------------------------

CAIS EDITORIAL BOARD

Michel Avital University of Amsterdam	Dinesh Batra Florida International University	Indranil Bose University of Hong Kong	Ashley Bush Florida State University
Fred Davis University of Arkansas, Fayetteville	Evan Duggan University of the West Indies	Ali Farhoomand University of Hong Kong	Sy Goodman Georgia Institute of Technology
Mary Granger George Washington University	Ake Gronlund University of Umea	Douglas Havelka Miami University	K.D. Joshi Washington State University
Chuck Kacmar University of Alabama	Michel Kalika University of Paris Dauphine	Julie Kendall Rutgers University	Claudia Loebbecke University of Cologne
Paul Benjamin Lowry Brigham Young University	Sal March Vanderbilt University	Don McCubbrey University of Denver	Fred Niederman St. Louis University
Shan Ling Pan National University of Singapore	Jackie Rees Purdue University	Jia-Lang Seng National Chengchi University	Paul Tallon Loyola College, Maryland
Thompson Teo National University of Singapore	Craig Tyran Western Washington University	Chelley Vician Michigan Technological University	Rolf Wigand University of Arkansas, Little Rock
Vance Wilson University of Toledo	Peter Wolcott University of Nebraska at Omaha	Yajiong Xue East Carolina University	

DEPARTMENTS

Global Diffusion of the Internet. Editors: Peter Wolcott and Sy Goodman	Information Technology and Systems. Editors: Sal March and Dinesh Batra
Papers in French Editor: Michel Kalika	Information Systems and Healthcare Editor: Vance Wilson

ADMINISTRATIVE PERSONNEL

James P. Tinsley AIS Executive Director	Vipin Arora CAIS Managing Editor University of Nebraska at Omaha	Copyediting by Carlisle Publishing Services
--------------------------------------------	------------------------------------------------------------------------	---------------------------------------------

